

**数据库实验报告**

**老 师： 王新梅**

**班 级： 220211**

**学 院： 未来技术学院**

**学 号： 20211003337**

**姓 名： 曾康慧**

目录

[实验一 实验报告 5](#_Toc154754137)

[一、 实验目的与要求 5](#_Toc154754138)

[二、 实验代码 5](#_Toc154754139)

[2.1 创建基本表 5](#_Toc154754140)

[2.2 单表查询 5](#_Toc154754141)

[2.3 分组统计查询 6](#_Toc154754142)

[2.4 连接查询（自身连接） 6](#_Toc154754143)

[2.5 连接查询（多表连接） 6](#_Toc154754144)

[三、 运行结果 6](#_Toc154754145)

[3.1创建基本表 6](#_Toc154754146)

[3.2 单表查询 8](#_Toc154754147)

[3.3 分组统计查询 9](#_Toc154754148)

[3.4 连接查询（自身连接） 10](#_Toc154754149)

[3.5 连接查询（多表连接） 11](#_Toc154754150)

[四、实验总结 11](#_Toc154754151)

[五、问题解决 11](#_Toc154754152)

[5.1 问题一 11](#_Toc154754153)

[实验二 实验报告 12](#_Toc154754154)

[一、 实验目的与要求 12](#_Toc154754155)

[二、 实验代码 12](#_Toc154754156)

[2.1 带有IN谓词的子查询 12](#_Toc154754157)

[2.2 带有比较运算符的子查询 12](#_Toc154754158)

[2.3 带有ANY（SOME）或ALL谓词的子查询 12](#_Toc154754159)

[2.4 带有EXISTS的子查询 12](#_Toc154754160)

[2.5集合查询 13](#_Toc154754161)

[三、 运行结果 13](#_Toc154754162)

[3.1 带有IN谓词的子查询 13](#_Toc154754163)

[3.2 带有比较运算符的子查询 13](#_Toc154754164)

[3.3 带有ANY（SOME）或ALL谓词的子查询 14](#_Toc154754165)

[3.4 带有EXISTS的子查询 14](#_Toc154754166)

[3.5 集合查询 14](#_Toc154754167)

[四、 实验总结 15](#_Toc154754168)

[五、 问题解决 15](#_Toc154754169)

[5.1 问题一 15](#_Toc154754170)

[实验三 实验报告 16](#_Toc154754171)

[一、 实验目的与要求 16](#_Toc154754172)

[二、实验代码 16](#_Toc154754173)

[2.1 插入一个元组 16](#_Toc154754174)

[2.2 插入子查询结果 16](#_Toc154754175)

[2.3 修改某元组的值 16](#_Toc154754176)

[2.4 修改多个元组的值 17](#_Toc154754177)

[2.5 带子查询的修改 17](#_Toc154754178)

[2.6 删除多个元组的值 17](#_Toc154754179)

[2.7 带子查询的删除 17](#_Toc154754180)

[三、运行结果 18](#_Toc154754181)

[3.1 插入一个元组 18](#_Toc154754182)

[3.2 插入子查询结果 18](#_Toc154754183)

[3.3 修改某元组的值 19](#_Toc154754184)

[3.4 修改多个元组的值 19](#_Toc154754185)

[3.5 带子查询的修改 20](#_Toc154754186)

[3.6 删除多个元组的值 20](#_Toc154754187)

[3.7 带子查询的删除 21](#_Toc154754188)

[四、实验总结 21](#_Toc154754189)

[五、问题解决 21](#_Toc154754190)

[5.1 问题一 21](#_Toc154754191)

[实验四 实验报告 22](#_Toc154754192)

[一、 实验目的与要求 22](#_Toc154754193)

[二、实验代码 22](#_Toc154754194)

[2.1 创建视图 22](#_Toc154754195)

[2.2 删除视图 22](#_Toc154754196)

[2.3 查询视图 22](#_Toc154754197)

[2.4 对非行列子集视图的查询 22](#_Toc154754198)

[三、运行结果 23](#_Toc154754199)

[3.1 创建视图 23](#_Toc154754200)

[3.2 删除视图 23](#_Toc154754201)

[3.3 查询视图 24](#_Toc154754202)

[3.4 对非行列子集视图的查询 24](#_Toc154754203)

[四、实验总结 24](#_Toc154754204)

[五、问题解决 25](#_Toc154754205)

[5.1 问题一 25](#_Toc154754206)

[实验五 实验报告 26](#_Toc154754207)

[一、 实验目的与要求 26](#_Toc154754208)

[二、实验代码 26](#_Toc154754209)

[2.1 系统主界面(MainWindow) 26](#_Toc154754210)

[2.2 登录界面(load) 39](#_Toc154754211)

[2.3 学生信息查询界面 40](#_Toc154754212)

[2.4 选课信息查询界面 41](#_Toc154754213)

[2.5 学生操作界面 42](#_Toc154754214)

[2.6 选课操作界面 44](#_Toc154754215)

[2.7 成绩统计界面 46](#_Toc154754216)

[三、运行结果 47](#_Toc154754217)

[3.1 系统主界面(MainWindow) 47](#_Toc154754218)

[3.2 登录界面(load) 49](#_Toc154754219)

[3.3 学生信息查询界面 50](#_Toc154754220)

[3.4 选课信息查询界面 51](#_Toc154754221)

[3.5 学生操作界面 52](#_Toc154754222)

[3.6 选课操作界面 56](#_Toc154754223)

[3.7 成绩统计界面 58](#_Toc154754224)

[3.8课程操作界面 60](#_Toc154754225)

[四、实验总结 64](#_Toc154754226)

[五、问题解决 64](#_Toc154754227)

[5.1 问题一 64](#_Toc154754228)

[5.2 问题二 64](#_Toc154754229)

[5.3 问题三 65](#_Toc154754230)

# 实验一 实验报告

## 实验目的与要求

(1)掌握SQL程序设计基本规范，熟练运用SQL语言实现数据定义、基本查询，包括单表查询、分组统计查询、连接查询

(2)能按照SQL程序设计规范写出具体的SQL查询语句，并调试通过

(3)支撑课程目标②④

## 实验代码

### 2.1 创建基本表

create table student

(sno CHAR(9) PRIMARY KEY,

sname CHAR(20) UNIQUE,

ssex CHAR(2),

sage SMALLINT,

sdept CHAR(20));

create table course

(cno CHAR(4) PRIMARY KEY,

cname CHAR(40) NOT NULL,

cpno CHAR(4),

ccredit SMALLUNT,

FOREIGN KEY(cpno) REFERENCES course(cno));

create table sc

(sno CHAR(9),

cno char(4),

grade smallint,

primary key(sno ,cno),

foreign key(sno) references student(sno),

foreign key(cno) references course(cno));

### 2.2 单表查询

select \*

from student;

### 2.3 分组统计查询

select Cno,count(Sno)

from sc

group by Cno;

### 2.4 连接查询（自身连接）

select f1.Cno,f2.Cno

from course f1,course f2

where f1.Cpno=f2.Cpno;

### 2.5 连接查询（多表连接）

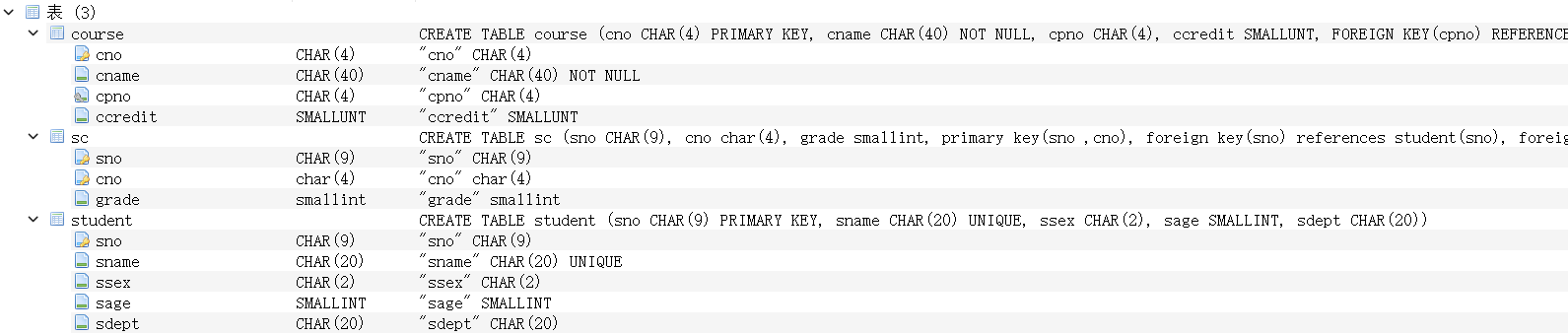
select student.Sno,Sname,Cname,grade

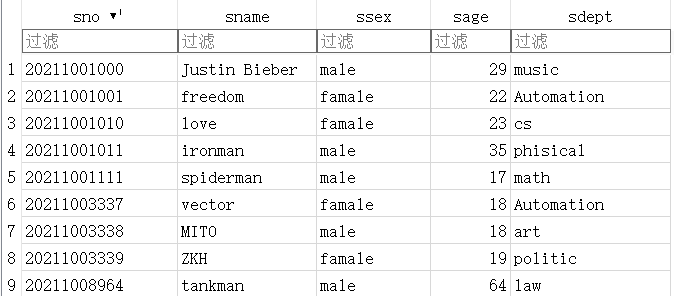
from student,sc,course

where student.Sno=sc.Sno AND sc.Cno=course.Cno;

## 运行结果

### 3.1创建基本表

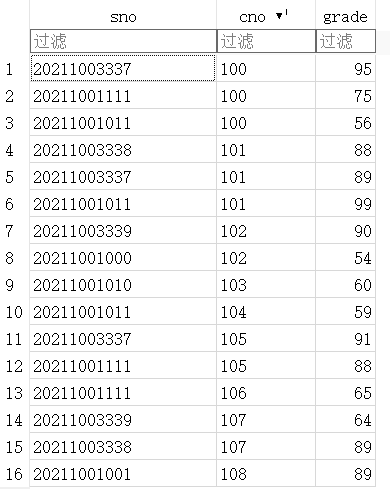




Student表



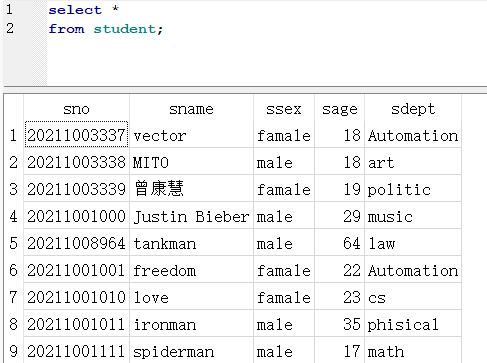
Course表



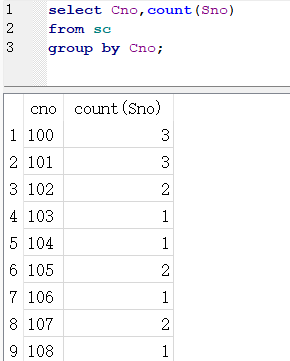
Sc表

### 3.2 单表查询

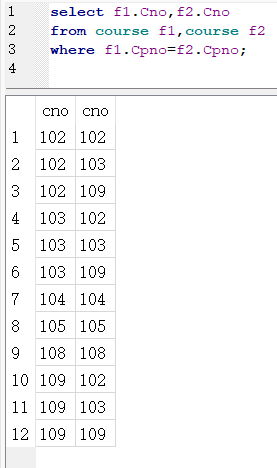
单表查询结果如下：



### 3.3 分组统计查询



### 3.4 连接查询（自身连接）



### 3.5 连接查询（多表连接）



## 四、实验总结

在本阶段的实习中，我掌握了基本表的建立、单表查询、分组统计查询、连接查询的使用方法。构建了sqlite3的环境，对数据库编程有了初步的认识。总体来说，实验一内容是比较简单的。

## 五、问题解决

### 5.1 问题一

创建基本表的过程中出现报错Error: table course already exists

原因是该表在之前已经被创建，无需再次创建。

# 实验二 实验报告

## 实验目的与要求

(1)掌握SQL嵌套查询和集合查询等各种高级查询的设计方法等。

(2)能按照SQL程序设计规范写出具体的SQL查询语句，并调试通过。

(3)支撑课程目标②。

## 实验代码

### 2.1 带有IN谓词的子查询

select Sno,Sname,Sdept

from student

where Sdept IN(select Sdept

from student where Sname=’曾康慧’);

### 2.2 带有比较运算符的子查询

select Sno,Sname,Sdept

from student

where Sdept=(select Sdept from student where Sname=**’**曾康慧’);

### 2.3 带有ANY（SOME）或ALL谓词的子查询

select Sname,Sage

from student

where Sage <(select MIN(Sage) from student where Sdept='cs')

AND Sdept <>'cs';

### 2.4 带有EXISTS的子查询

select Sname

from student

where exists(select \* from sc where Sno=student.Sno AND Cno='100');

### 2.5集合查询

select Sno

from sc

where Cno='101'

UNION

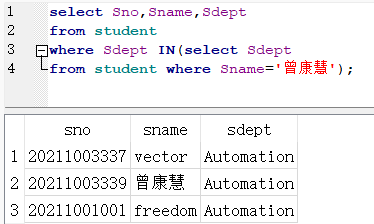
select Sno

from sc

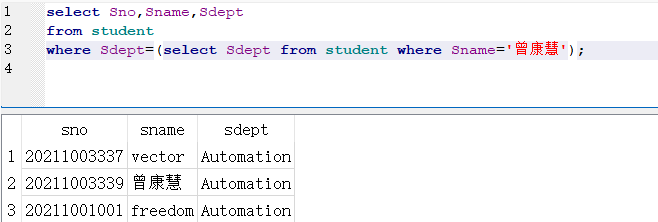
where Cno='102';

## 运行结果

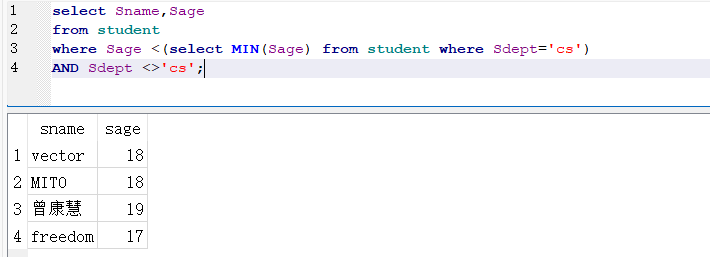
### 3.1 带有IN谓词的子查询



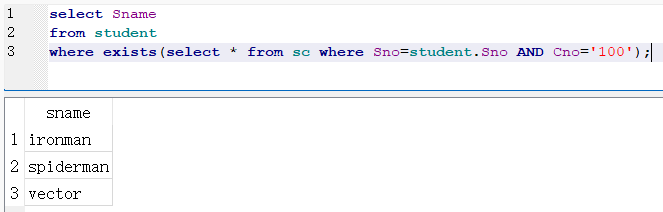
### 3.2 带有比较运算符的子查询



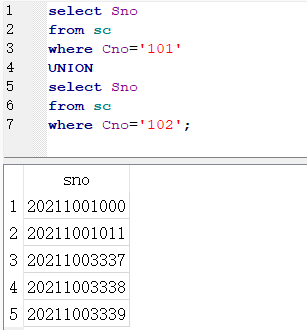
### 3.3 带有ANY（SOME）或ALL谓词的子查询



### 3.4 带有EXISTS的子查询



### 3.5 集合查询



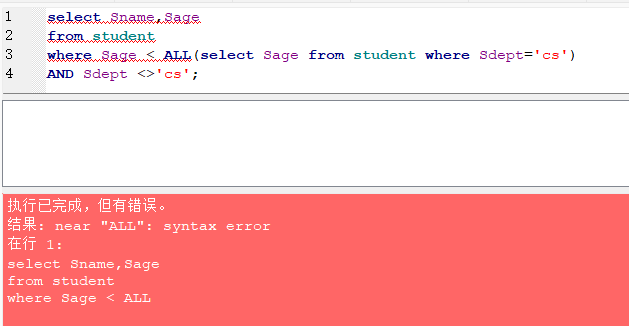
## 实验总结

在本阶段的实验中，我掌握了带有IN的子查询、带有比较运算符的子查询、带有ANY谓词的子查询、带有exists的子查询以及集合查询。在实验过程中，我发现了SQlite中部分SQL语句的用法与我们在课本上所学的SQL语句是不一样的。

## 问题解决

### 5.1 问题一

使用含有ALL的查询时，程序报错，见下图。



原因是ALL的语法再SQlite中与一般的SQL语句有所不同，我们可以用<MIN()等类似的查询方式代替ALL在SQL语句中的作用。

# 实验三 实验报告

## 实验目的与要求

(1)熟悉数据库的数据更新操作，能够使用SQL语句对数据库进行数据的插入、修改、删除。

(2)理解和掌握insert、update和delete语法结构的各个组成成分，结合嵌套SQL子查询，分别设计几种不同形式的插入、修改和删除数据的语句，并调试通过。

(3)支撑课程目标②④。

## 二、实验代码

### 2.1 插入一个元组

insert

into sc(Sno,Cno)

values('20211003337','102');

### 2.2 插入子查询结果

create table Dept\_age

(Sdept CHAR(15),

Avg\_age smallint);

insert

into Dept\_age(Sdept,Avg\_age)

select Sdept,AVG(Sage)

from student

group by Sdept;

### 2.3 修改某元组的值

update student

set Sage=25

where Sno='20211001010';

### 2.4 修改多个元组的值

update student

set Sage=Sage+1;

select \* from student;

### 2.5 带子查询的修改

update sc

 set grade=0

where Sno IN

(select Sno

from student

where Sdept='cs');

### 2.6 删除多个元组的值

delete

from sc

where Sno = '20211003337';

### 2.7 带子查询的删除

delete

from sc

where Sno IN

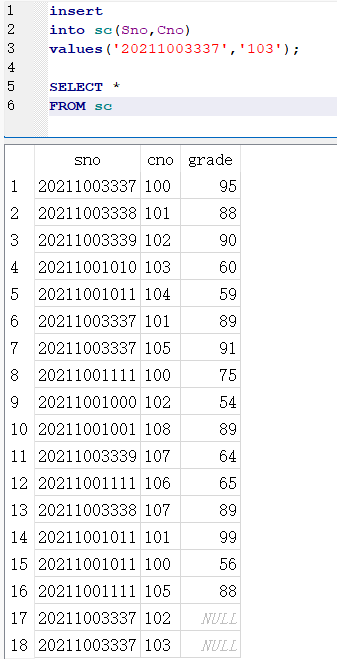
(select Sno

from student

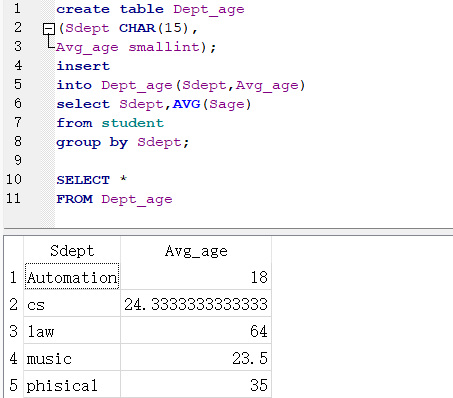
where Sdept='cs');

## 三、运行结果

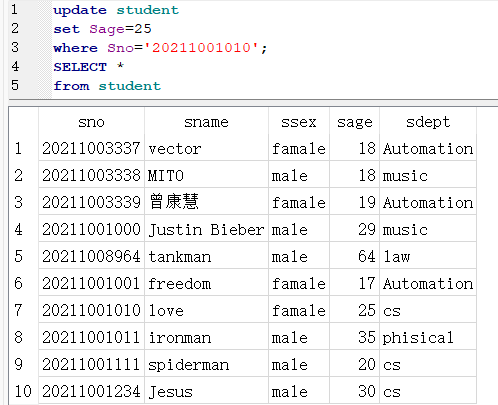
### 3.1 插入一个元组



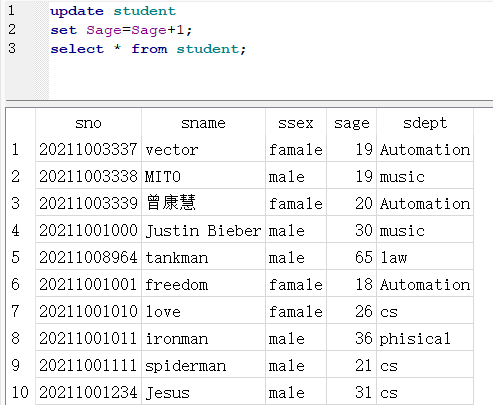
### 3.2 插入子查询结果



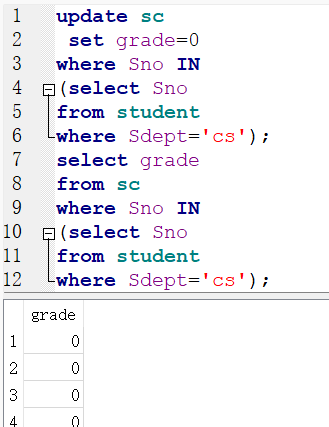
### 3.3 修改某元组的值



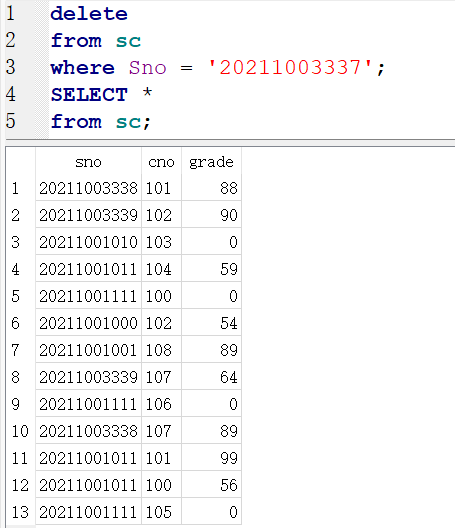
### 3.4 修改多个元组的值



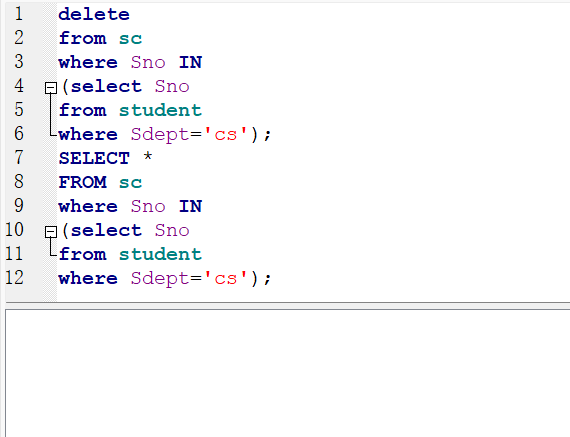
### 3.5 带子查询的修改



### 3.6 删除多个元组的值



### 3.7 带子查询的删除



## 四、实验总结

在该部分实验中，我掌握了insert、update、delete等语句的用法，对SQL语句的理解有了进一步的提升。在使用insert、update、delete等语句时需要时刻注意，不能破坏实体完整性、参照完整性、用户定义完整性。

## 五、问题解决

### 5.1 问题一

问题描述：在SQLite中使用inse、update、delete语句时，系统没有自动进行完整性定义检查。

解决方法：可以通过自定义用户定义完整性、设置断言等方法来解决这一问题。

# 实验四 实验报告

## 实验目的与要求

(1)熟悉SQL语言有关视图的操作，能够熟练使用SQL语句来创建需要的视图，定义数据库外模式，并能使用所创建的视图实现数据管理

(2)理解掌握视图消解执行原理，掌握可更新视图和不可更新视图的区别

(3)支撑课程目标②④

## 二、实验代码

### 2.1 创建视图

create view is\_student

as

select Sno,Sname,Sage

from student

where Sdept='Automation';

### 2.2 删除视图

drop view is\_student;

### 2.3 查询视图

create view cs

as

select sno,sname,sage

from student

where sdept='cs';

### 2.4 对非行列子集视图的查询

create view average\_view (sno,gavg)

as

elect sno,AVG(grade)

from sc

group by sno;

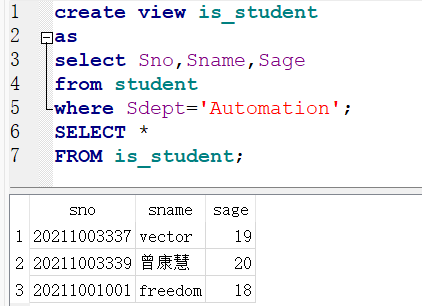
select \*

from average\_view

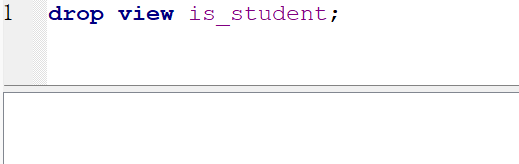
where gavg>80;

## 三、运行结果

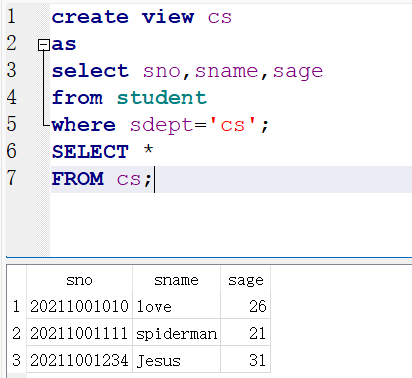
### 3.1 创建视图



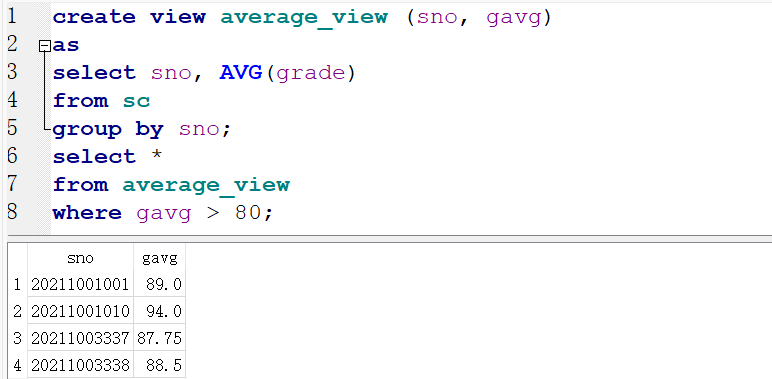
### 3.2 删除视图



### 3.3 查询视图



### 3.4 对非行列子集视图的查询



## 四、实验总结

在该部分实验中，我掌握了视图创建、更新视图、视图消解的方法。在实验过程中我发现在sqlite中部分非行列子集视图也是可以正常消解的。

在创建视图时，子查询中的属性值出错并不会立马报错，而是在视图被调用的时候才会出现报错。这说明了视图建立后并不会立马被执行，而是在被调用的时候才会执行。进而说明了视图只是一个虚表，并不是实际存在的表。

## 五、问题解决

### 5.1 问题一

问题描述：创建视图时出现报错。

问题原因：在创建视图时，属性名输入错误, "average"是SQL中的保留字，可能会导致问题。

解决方法：将视图删除，重新创建。



# 实验五 实验报告

## 实验目的与要求

(1)通过实验，掌握数据库设计基本方法，熟练应用高级语言结合SQL语言完成与信息系统相关的数据管理

(2)支撑课程目标①~④

## 二、实验代码

### 2.1 系统主界面(MainWindow)

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QMessageBox>

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

if(flag\_log=="0"){

ui->find\_stu\_all->setEnabled(false);

ui->find\_stu->setEnabled(false);

ui->find\_stu\_all\_2->setEnabled(false);

ui->find\_sc->setEnabled(false);

ui->find\_sc\_2->setEnabled(false);

ui->find\_sc\_3->setEnabled(false);

ui->find\_sc\_4->setEnabled(false);

ui->grade\_sel->setEnabled(false);

ui->label\_2->setText("请登录！");

}

sql\_connect();

connect(find,&Find::send\_sno,this,&MainWindow::rec\_sno);

connect(find,&Find::send\_sname,this,&MainWindow::rec\_sname);

connect(find\_sc,&Find\_sc::send\_sno,this,&MainWindow::rec\_sno\_sc); connect(find\_sc,&Find\_sc::send\_sname,this,&MainWindow::rec\_sname\_sc);

connect(stuset,&stu\_set::new\_student,this,&MainWindow::rec\_new\_stu); connect(stuset,&stu\_set::del\_student,this,&MainWindow::rec\_del\_stu); connect(stuset,&stu\_set::rename\_stu,this,&MainWindow::rec\_rename\_stu); connect(stuset,&stu\_set::reage\_stu,this,&MainWindow::rec\_reage\_stu); connect(stuset,&stu\_set::resex\_stu,this,&MainWindow::rec\_resex\_stu); connect(stuset,&stu\_set::redept\_stu,this,&MainWindow::rec\_redept\_stu); connect(stuset,&stu\_set::recity\_stu,this,&MainWindow::rec\_recity\_stu); connect(coset,&course\_set::new\_course,this,&MainWindow::rec\_new\_course); connect(coset,&course\_set::del\_course,this,&MainWindow::rec\_del\_course); connect(coset,&course\_set::rcname\_course,this,&MainWindow::rec\_recname\_course); connect(coset,&course\_set::rcpno\_course,this,&MainWindow::rec\_recpno\_course);

connect(coset,&course\_set::rcredit\_course,this,&MainWindow::rec\_recredit\_course);

connect(coset,&course\_set::find\_cno,this,&MainWindow::find\_cno);

connect(coset,&course\_set::find\_cname,this,&MainWindow::find\_cname);

connect(scset,&sc\_set::new\_sc,this,&MainWindow::rec\_new\_sc);

connect(scset,&sc\_set::del\_sc,this,&MainWindow::rec\_del\_sc);

connect(scset,&sc\_set::cha\_sc,this,&MainWindow::rec\_cha\_sc);

connect(gsel,&grade\_sel::stu\_sel,this,&MainWindow::rec\_stu\_sel);

connect(gsel,&grade\_sel::course\_sel,this,&MainWindow::rec\_course\_sel);

connect(gsel,&grade\_sel::grade\_stat,this,&MainWindow::rec\_grade\_stat);

connect(ld,&load::man\_mode,this,&MainWindow::rec\_man\_mode);

}

MainWindow::~***MainWindow***()

{

delete ui;

}

bool MainWindow::**sql\_connect**()

{

QSqlDatabase db = QSqlDatabase::addDatabase("QSQLITE");

db.setDatabaseName("student.db");

if (!db.open()) {

QMessageBox::critical(nullptr, QObject::tr("Cannot open database"),

QObject::tr("Unable to establish a database connection.\n"

"This example needs SQLite support. Please read "

"the Qt SQL driver documentation for information how "

"to build it.\n\n"

"Click Cancel to exit."), QMessageBox::Cancel);

return false;

}

return true;

}

void MainWindow::**rec\_sno**(QString m\_sno)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("select \* from student where sno = ? ");

qr.addBindValue(m\_sno);//有点类似格式化字符串的作用

qr.exec();

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_sname**(QString m\_sname)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr；

qr.prepare("select \* from student where sname = ? ");

qr.addBindValue(m\_sname);//有点类似格式化字符串的作用

qr.exec();

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_sno\_sc**(QString sno)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("select sc.sno,sname,scity,cname,grade from sc,student,course where sc.sno=student.sno AND sc.sno=? AND course.cno=sc.cno");

qr.addBindValue(sno);//有点类似格式化字符串的作用

qr.exec();

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_sname\_sc**(QString sname)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("select sc.sno,sname,scity,cname,grade from sc,student,course where sc.sno=student.sno AND sname=? AND course.cno=sc.cno");

qr.addBindValue(sname);//有点类似格式化字符串的作用

qr.exec();

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_new\_stu**(QString sno, QString sname, QString ssex, QString sage, QString sdept,QString scity)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("insert into student(Sno,Sname,Ssex,Sage,Sdept,Scity) values(:a,:b,:c,:d,:e,:f)");

qr.bindValue(":a",sno);//有点类似格式化字符串的作用

qr.bindValue(":b",sname);

qr.bindValue(":c",ssex);

qr.bindValue(":d",sage);

qr.bindValue(":e",sdept);

qr.bindValue(":f",scity)；

qr.exec();

qr.exec("select \* from student order by sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_del\_stu**(QString sno)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("delete from student where sno = ?");

qr.addBindValue(sno);//有点类似格式化字符串的作用

qr.exec();

qr.exec("select \* from student order by sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_rename\_stu**(QString sno, QString sname)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update student set Sname = :b where Sno = :a");

qr.bindValue(":a",sno);//有点类似格式化字符串的作用

qr.bindValue(":b",sname);

qr.exec();

qr.exec("select \* from student order by sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_reage\_stu**(QString sno, QString sage)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update student set Sage = :b where Sno = :a");

qr.bindValue(":a",sno);//有点类似格式化字符串的作用

qr.bindValue(":b",sage);

qr.exec();

qr.exec("select \* from student order by sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_resex\_stu**(QString sno, QString sex)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update student set Ssex = :b where Sno = :a");

qr.bindValue(":a",sno);//有点类似格式化字符串的作用

qr.bindValue(":b",sex);

qr.exec();

qr.exec("select \* from student order by sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_redept\_stu**(QString sno, QString sdept)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update student set Sdept = :b where Sno = :a");

qr.bindValue(":a",sno);//有点类似格式化字符串的作用

qr.bindValue(":b",sdept);

qr.exec();

qr.exec("select \* from student order by sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_recity\_stu**(QString sno, QString scity)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update student set Scity = :b where Sno = :a");

qr.bindValue(":a",sno);//有点类似格式化字符串的作用

qr.bindValue(":b",scity);

qr.exec();

qr.exec("select \* from student order by sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_new\_course**(QString cno, QString cname, QString cpno, QString credit)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("insert into course(Cno,Cname,Cpno,Ccredit) values(:a,:b,:c,:d)");

qr.bindValue(":a",cno);//有点类似格式化字符串的作用

qr.bindValue(":b",cname);

qr.bindValue(":c",cpno);

qr.bindValue(":d",credit);

qr.exec();

qr.exec("select \* from course order by cno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_del\_course**(QString cno)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("delete from course where Cno = ?");

qr.addBindValue(cno);//有点类似格式化字符串的作用

qr.exec();

qr.exec("select \* from course order by cno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_recname\_course**(QString cno, QString cname)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update course set Cname = :b where Cno = :a");

qr.bindValue(":a",cno);//有点类似格式化字符串的作用

qr.bindValue(":b",cname);

qr.exec();

qr.exec("select \* from course order by cno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_recpno\_course**(QString cno, QString cpno)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update course set Cpno = :b where Cno = :a");

qr.bindValue(":a",cno);//有点类似格式化字符串的作用

qr.bindValue(":b",cpno);

qr.exec();

qr.exec("select \* from course order by cno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_recredit\_course**(QString cno, QString credit)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update course set Ccredit = :b where Cno = :a");

qr.bindValue(":a",cno);//有点类似格式化字符串的作用

qr.bindValue(":b",credit);

qr.exec();

qr.exec("select \* from course order by cno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_new\_sc**(QString sno, QString cno, QString grade)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("insert into sc(Sno,Cno,grade) values(:a,:b,:c)");

qr.bindValue(":a",sno);//有点类似格式化字符串的作用

qr.bindValue(":b",cno);

qr.bindValue(":c",grade);

qr.exec();

qr.exec("select sc.Sno,Sname,sc.Cno,Cname,grade from student,sc,course where sc.Sno=student.Sno AND sc.Cno=course.Cno order by sc.Sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_del\_sc**(QString sno, QString cno)

{

//QMessageBox::question(NULL, "提示", "确定要删除吗？", QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("delete from sc where Cno = :a AND Sno = :b");

qr.bindValue(":a",cno);//有点类似格式化字符串的作用

qr.bindValue(":b",sno);

qr.exec();

qr.exec("select sc.Sno,Sname,sc.Cno,Cname,grade from student,sc,course where sc.Sno=student.Sno AND sc.Cno=course.Cno order by sc.Sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_cha\_sc**(QString sno, QString cno, QString grade)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("update sc set grade = :b where Cno = :a AND Sno = :c");

qr.bindValue(":a",cno);//有点类似格式化字符串的作用

qr.bindValue(":b",grade);

qr.bindValue(":c",sno);

qr.exec();

qr.exec("select sc.Sno,Sname,sc.Cno,Cname,grade from student,sc,course where sc.Sno=student.Sno AND sc.Cno=course.Cno order by sc.Sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_stu\_sel**(QString sno)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("select student.Sno,Sname,AVG(grade) 平均成绩,MAX(grade) 最高分,MIN(grade) 最低分 from sc,student,course where sc.Sno=student.Sno AND sc.Cno=course.Cno AND sc.Sno = ? group by sc.Sno");

qr.addBindValue(sno);//有点类似格式化字符串的作用

qr.exec();

//qr.exec("select \* from course");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_course\_sel**(QString cno)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("select sc.Cno,Cname,AVG(grade) 平均成绩,MAX(grade) 最高分,MIN(grade) 最低分 from sc,student,course where sc.Sno=student.Sno AND sc.Cno=course.Cno AND sc.Cno = ? group by sc.Cno");

qr.addBindValue(cno);//有点类似格式化字符串的作用

qr.exec();

//qr.exec("select \* from course");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_man\_mode**(QString)

{

if(flag\_log=="0"){

ui->find\_stu\_all->setEnabled(false);

ui->find\_stu->setEnabled(false);

ui->find\_stu\_all\_2->setEnabled(false);

ui->find\_sc->setEnabled(false);

ui->find\_sc\_2->setEnabled(false);

ui->find\_sc\_3->setEnabled(false);

ui->find\_sc\_4->setEnabled(false);

ui->grade\_sel->setEnabled(false);

ui->label\_2->setText("请登录！");

}

else if(flag\_log=="1"){

ui->find\_stu\_all->setEnabled(true);

ui->find\_stu->setEnabled(true);

ui->find\_stu\_all\_2->setEnabled(true);

ui->find\_sc->setEnabled(true);

ui->grade\_sel->show();

ui->find\_stu\_all->show();

ui->find\_stu\_all\_2->show();

ui->find\_sc\_3->show();

ui->find\_sc\_4->show();

ui->label\_2->setText("老师您好！");

ui->find\_sc\_2->setEnabled(true);

ui->find\_sc\_3->setEnabled(true);

ui->find\_sc\_4->setEnabled(true);

ui->grade\_sel->setEnabled(true);

ld->close();

}

else if(flag\_log=="2"){

ui->find\_stu->setEnabled(true);

ui->find\_sc->setEnabled(true);

ui->find\_sc\_2->hide();

ui->grade\_sel->hide();

ui->find\_stu\_all->hide();

ui->find\_stu\_all\_2->hide();

ui->find\_sc\_3->hide();

ui->find\_sc\_4->hide();

ui->label\_2->setText("同学你好！");

ld->close();

}

}

void MainWindow::**find\_cno**(QString cno)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("select sc.Cno,cname,cpno,ccredit,count(sc.Cno) 选课人数 from course, sc where sc.Cno = ? AND sc.Cno=course.Cno");

qr.addBindValue(cno);//有点类似格式化字符串的作用

qr.exec();

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**find\_cname**(QString cname)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("select sc.Cno,cname,cpno,ccredit,count(sc.Cno) 选课人数 from course,sc where cname = ? AND sc.Cno=course.Cno");

qr.addBindValue(cname);//有点类似格式化字符串的作用

qr.exec();

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**rec\_grade\_stat**(QString g\_min, QString g\_max)

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.prepare("select sname,sc.cno,cname,grade from sc,student,course where student.sno=sc.sno AND course.cno=sc.cno AND grade > :a AND grade < :b");

qr.bindValue(":a",g\_min);//有点类似格式化字符串的作用

qr.bindValue(":b",g\_max);

qr.exec();

//qr.exec("select Sname,sc.Sno,sc.Cno,Cname,grade from student,sc,course where sc.Sno=student.Sno AND sc.Cno=course.Cno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**on\_find\_stu\_clicked**()

{

find->show();

}

void MainWindow::**on\_find\_stu\_all\_clicked**()

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.exec("select \* from student order by sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**on\_find\_stu\_all\_2\_clicked**()

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.exec("select student.sno,sname,cname,sc.cno,scity,grade from sc,student,course where student.sno=sc.sno AND course.cno=sc.cno order by student.sno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

}

void MainWindow::**on\_find\_sc\_clicked**()

{

find\_sc->show();

}

void MainWindow::**on\_find\_sc\_2\_clicked**()

{

stuset->show();

}

void MainWindow::**on\_find\_sc\_3\_clicked**()

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.exec("SELECT course.Cno, cname, cpno, ccredit, COUNT(sc.Cno) AS 选课人数 FROM course LEFT JOIN sc ON sc.Cno = course.Cno GROUP BY course.Cno, cname, cpno, ccredit;");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

coset->show();

}

void MainWindow::**on\_find\_sc\_4\_clicked**()

{

studentqrModel=new QSqlQueryModel(this);

sql\_connect();

QSqlQuery qr;

qr.exec("select Sname,sc.Sno,sc.Cno,Cname,scity,grade from student,sc,course where sc.Sno=student.Sno AND sc.Cno=course.Cno");

studentqrModel->setQuery(qr);

ui->tableView\_manage->*setModel*(studentqrModel);

scset->show();

}

void MainWindow::**on\_grade\_sel\_clicked**()

{

gsel->show();

}

void MainWindow::**on\_loadsys\_clicked**()

{

ld->show();

}

void MainWindow::**on\_ex\_clicked**()

{

QMessageBox::question(NULL, "提示", "确定要退出吗？", QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);

ui->find\_stu\_all->setEnabled(false);

ui->find\_stu->setEnabled(false);

ui->find\_stu\_all\_2->setEnabled(false);

ui->find\_sc->setEnabled(false);

ui->find\_sc\_2->setEnabled(false);

ui->find\_sc\_3->setEnabled(false);

ui->find\_sc\_4->setEnabled(false);

ui->grade\_sel->setEnabled(false);

flag\_log="0";

ld->show();

}

### 2.2 登录界面(load)

#include "load.h"

#include "ui\_load.h"

#include <QMessageBox>

QString flag\_log="0";

load::**load**(QWidget \*parent) :

QWidget(parent),

ui(new Ui::load)

{

ui->setupUi(this);

}

load::~***load***()

{

delete ui;

}

void load::**on\_pushButton\_clicked**()

{

QString uname;

QString password;

uname=ui->lineEdit->text();

password=ui->lineEdit\_2->text();

if(uname == "root" && password =="123"){

flag\_log="1";

}

else if(uname == "stu" && password =="123"){

flag\_log="2";

}

else{

flag\_log="0";

log->show();

}

emit man\_mode(flag\_log);

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

}

void load::**on\_exit\_clicked**()

{

exit(0);

}

### 2.3 学生信息查询界面

#include "find.h"

#include "ui\_find.h"

Find::**Find**(QWidget \*parent) :

QWidget(parent),

ui(new Ui::Find)

{

ui->setupUi(this);

}

Find::~***Find***()

{

delete ui;

}

void Find::**on\_find\_sure\_clicked**()

{

emit send\_sno(ui->input\_sno->text());

ui->input\_sno->clear();

}

void Find::**on\_find\_sure\_2\_clicked**()

{

emit send\_sname(ui->input\_sname->text());

ui->input\_sname->clear();

}

### 2.4 选课信息查询界面

#include "find\_sc.h"

#include "ui\_find\_sc.h"

Find\_sc::**Find\_sc**(QWidget \*parent) :

QWidget(*parent*),

ui(new Ui::Find\_sc)

{

ui->setupUi(this);

}

Find\_sc::~***Find\_sc***()

{

delete ui;

}

void Find\_sc::**on\_find\_sure\_3\_clicked**()

{

emit send\_sno(ui->input\_sno\_1->text());

ui->input\_sno\_1->clear();

}

void Find\_sc::**on\_find\_sure\_4\_clicked**()

{

emit send\_sname(ui->input\_sname\_1->text());

ui->input\_sname\_1->clear();

}

### 2.5 学生操作界面

#include "stu\_set.h"

#include "ui\_stu\_set.h"

#include <QMessageBox>

stu\_set::**stu\_set**(QWidget \*parent) :

QWidget(parent),

ui(new Ui::stu\_set)

{

ui->setupUi(this);

}

stu\_set::~***stu\_set***()

{

delete ui;

}

void stu\_set::**on\_new\_stu\_clicked**()

{

QString Sno;

QString Sname;

QString Ssex;

QString Sage;

QString Sdept;

QString Scity;

Sno=ui->lineEdit->text();

Sname=ui->lineEdit\_2->text();

Ssex=ui->lineEdit\_3->text();

Sage=ui->lineEdit\_4->text();

Sdept=ui->lineEdit\_5->text();

Scity=ui->lineEdit\_9->text();

emit new\_student(Sno,Sname,Ssex,Sage,Sdept,Scity);

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

ui->lineEdit\_3->clear();

ui->lineEdit\_4->clear();

ui->lineEdit\_5->clear();

ui->lineEdit\_9->clear();

}

void stu\_set::**on\_pushButton\_clicked**()

{

QMessageBox::question(NULL, "提示", "确定要删除吗？", QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);

emit del\_student(ui->lineEdit\_6->text());

ui->lineEdit\_6->clear();

}

void stu\_set::**on\_rename\_clicked**()

{

QString Sno;

QString Sname;

Sno=ui->lineEdit\_7->text();

Sname=ui->lineEdit\_8->text();

emit rename\_stu(Sno,Sname);

ui->lineEdit\_8->clear();

}

void stu\_set::**on\_reage\_clicked**()

{

QString Sno;

QString Sage;

Sno=ui->lineEdit\_7->text();

Sage=ui->lineEdit\_8->text();

emit reage\_stu(Sno,Sage);

ui->lineEdit\_8->clear();

}

void stu\_set::**on\_resex\_clicked**()

{

QString Sno;

QString Ssex;

Sno=ui->lineEdit\_7->text();

Ssex=ui->lineEdit\_8->text();

emit resex\_stu(Sno,Ssex);

ui->lineEdit\_8->clear();

}

void stu\_set::**on\_redept\_clicked**()

{

QString Sno;

QString Sdept;

Sno=ui->lineEdit\_7->text();

Sdept=ui->lineEdit\_8->text();

emit redept\_stu(Sno,Sdept);

ui->lineEdit\_8->clear();

}

void stu\_set::**on\_recity\_clicked**()

{

QString Sno;

QString Scity;

Sno=ui->lineEdit\_7->text();

Scity=ui->lineEdit\_8->text();

emit recity\_stu(Sno,Scity);

ui->lineEdit\_8->clear();

}

### 2.6 选课操作界面

#include "sc\_set.h"

#include "ui\_sc\_set.h"

#include <QMessageBox>

sc\_set::**sc\_set**(QWidget \*parent) :

QWidget(parent),

ui(new Ui::sc\_set)

{

ui->setupUi(this);

}

sc\_set::~***sc\_set***()

{

delete ui;

}

void sc\_set::**on\_new\_sc\_clicked**()

{

QString sno;

QString cno;

QString grade;

sno=ui->lineEdit->text();

cno=ui->lineEdit\_2->text();

grade=ui->lineEdit\_3->text();

emit new\_sc(sno,cno,grade);

ui->lineEdit->clear();

ui->lineEdit\_2->clear();

ui->lineEdit\_3->clear();

}

void sc\_set::**on\_del\_sc\_clicked**()

{

QMessageBox::question(NULL, "提示", "确定要删除吗？", QMessageBox::Yes | QMessageBox::No, QMessageBox::Yes);

QString sno;

QString cno;

sno=ui->lineEdit\_4->text();

cno=ui->lineEdit\_5->text();

emit del\_sc(sno,cno);

ui->lineEdit\_4->clear();

ui->lineEdit\_5->clear();

}

void sc\_set::**on\_cha\_sc\_clicked**()

{

QString sno;

QString cno;

QString grade;

sno=ui->lineEdit\_6->text();

cno=ui->lineEdit\_7->text();

grade=ui->lineEdit\_8->text();

emit cha\_sc(sno,cno,grade);//发送

ui->lineEdit\_6->clear();

ui->lineEdit\_7->clear();

ui->lineEdit\_8->clear();

}

### 2.7 成绩统计界面

#include "grade\_sel.h"

#include "ui\_grade\_sel.h"

grade\_sel::**grade\_sel**(QWidget \*parent) :

QWidget(parent),

ui(new Ui::grade\_sel)

{

ui->setupUi(this);

}

grade\_sel::~***grade\_sel***()

{

delete ui;

}

void grade\_sel::**on\_stu\_sel\_clicked**()

{

emit stu\_sel(ui->lineEdit->text());

ui->lineEdit->clear();

}

void grade\_sel::**on\_coure\_sel\_clicked**()

{

emit course\_sel(ui->lineEdit\_2->text());

ui->lineEdit\_2->clear();

}

void grade\_sel::**on\_grade\_stat\_clicked**()

{

QString g\_min;

QString g\_max;

g\_min = ui->lineEdit\_3->text();

g\_max = ui->lineEdit\_4->text();

emit grade\_stat(g\_min,g\_max);

ui->lineEdit\_3->clear();

ui->lineEdit\_4->clear();

}

## 三、运行结果

### 3.1 系统主界面(MainWindow)

系统主界面如下图所示。



图 1 系统主界面（未登录）

该界面为未登录时所看到的界面，教师有该系统的最高权限。学生登录系统后会有部分按键变暗无法使用，如下图所示：



图 2 系统主界面（学生端）



图 3 系统主界面（教室端）

### 3.2 登录界面(load)

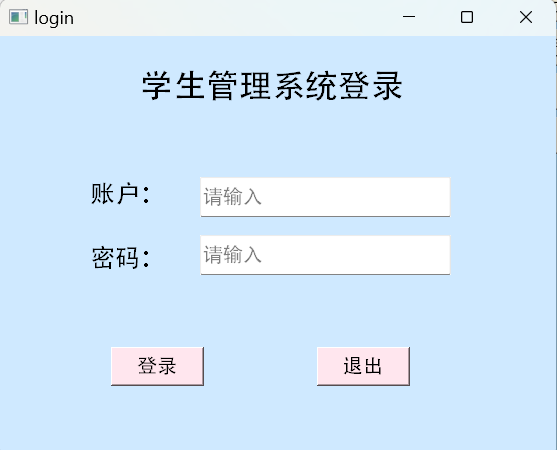


图 4 登录界面

一共有两类账号，一类是教师账号，一类是学生账号，两类账号登录会有不同的界面。

如果账号密码错误，则提示：

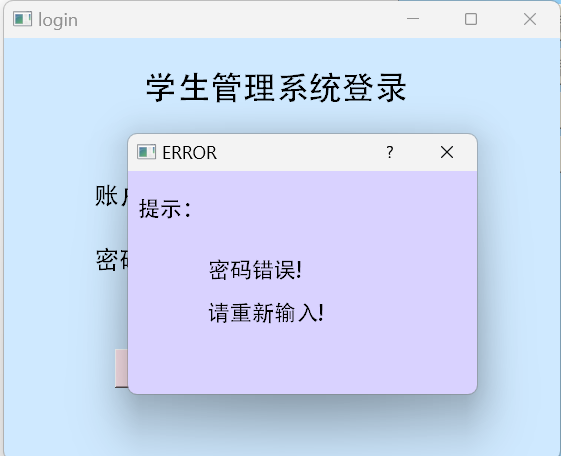


图 5 密码错误提示

### 3.3 学生信息查询界面

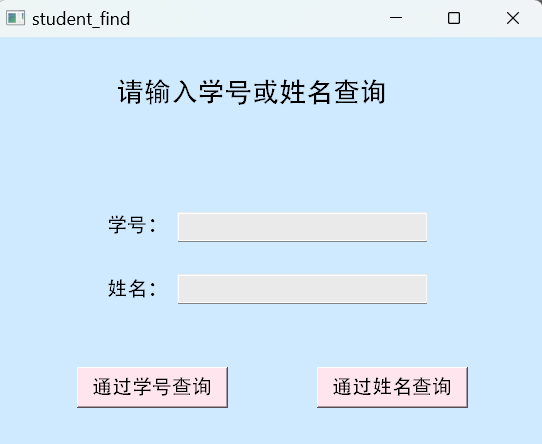


图 6 学生信息查询界面

可通过学号或姓名查询。

查询结果得：



图 7 学生信息查询结果

### 3.4 选课信息查询界面

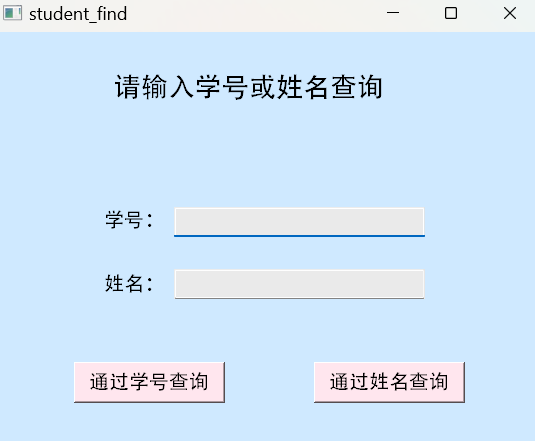


图 8 选课信息查询界面

可通过学号或姓名查询。

查询结果得：



图 9 选课信息查询结果

### 3.5 学生操作界面

显示所有学生信息如下：



图 10 显示所有学生信息界面

新增学生界面：

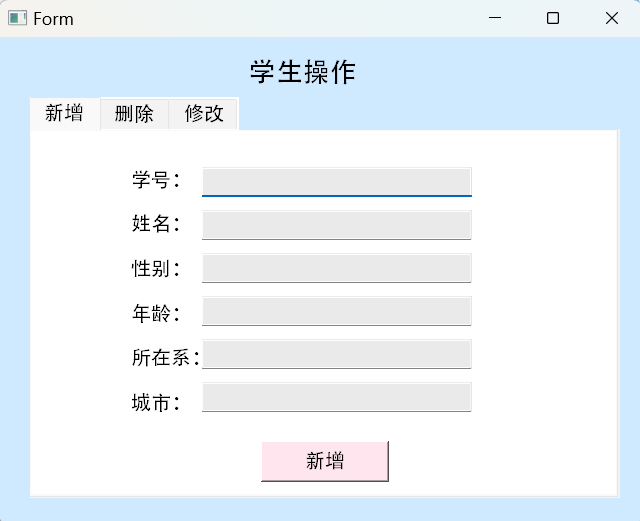


图 11 学生操作新增界面

新增学生完颜慧德，结果如下：



图 12 新增学生结果

删除学生界面：

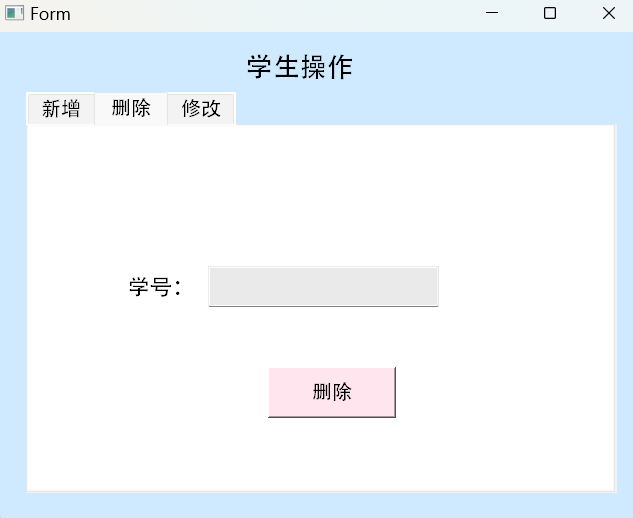


图 13 学生操作删除界面

删除时会跳出“确定要删除吗”的提示窗口：

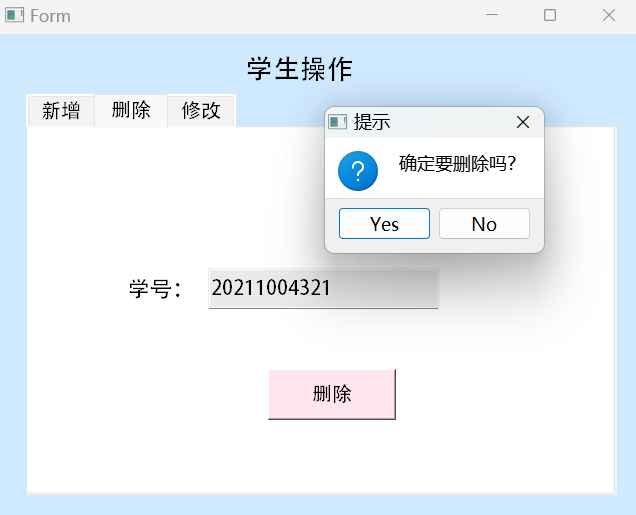


图 14 提示窗口界面

删除结果得：



图 15 删除结果

修改学生信息表界面，此处修改曾康慧同学的城市为孝感：



图 16 学生操作修改界面

修改结果得：



图 17 学生信息修改结果

### 3.6 选课操作界面

显示全部的学生选课信息如下;



图 18 显示全部学生的选课信息界面

选课界面有新增、删除、修改三个字界面

新增选课界面如下：

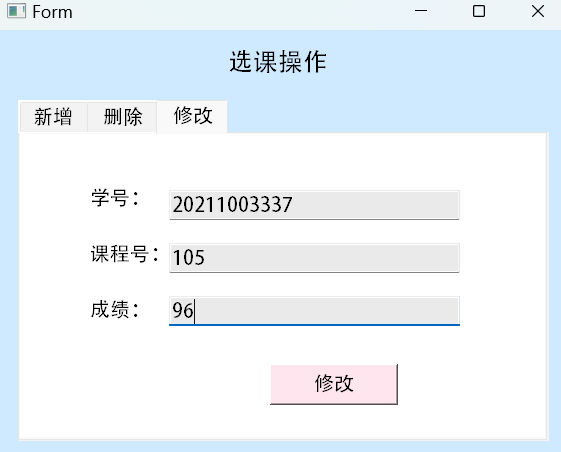


图 19 选课界面新增界面

新增选课结果如下，vector同学多了一门嵌入式原理的选课成绩：



图 20 新增选课结果

删除选课界面如下：

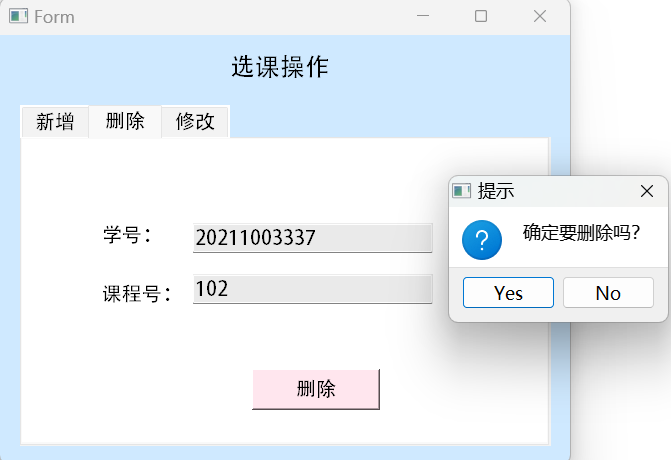


图 21 选课界面删除界面

删除vector同学的数据库选课，结果如下：



图 22 选课删除结果

修改选课信息界面如下：

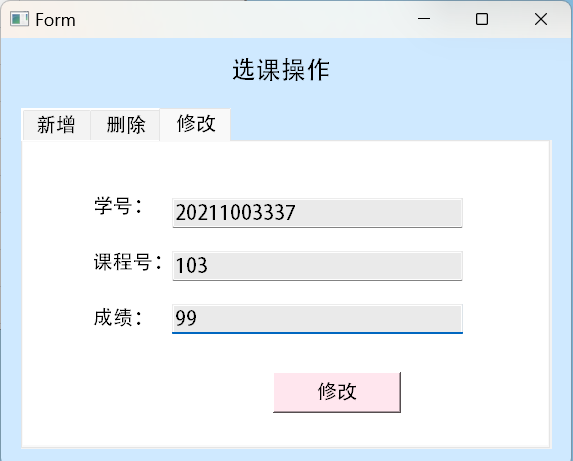


图 23 修改选课界面

vector同学的103课程自动控制原理的分数改为了99分：

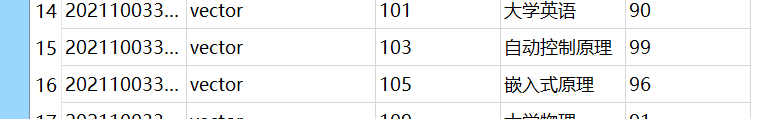


图 24 删除选课结果

### 3.7 成绩统计界面

学生个人成绩统计：

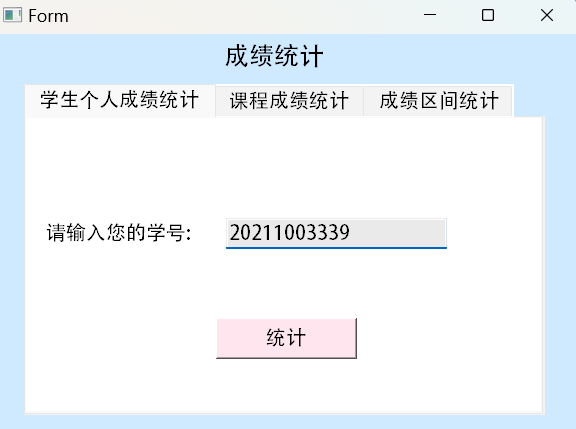


图 25 成绩统计界面学生个人成绩统计

曾康慧同学的个人成绩统计结果如下：



图 26学生个人成绩统计结果

课程成绩统计界面:

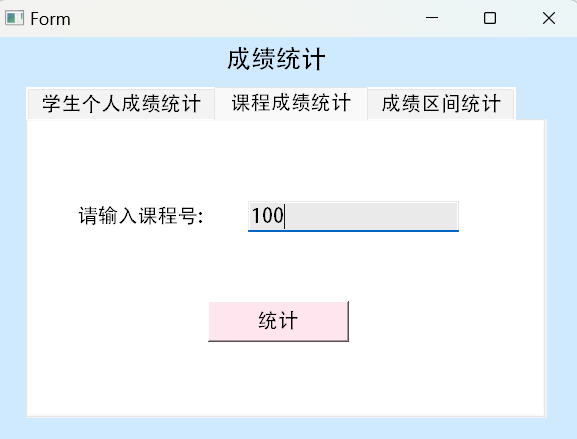


图 27 成绩统计界面课程成绩统计

课程号为100的高等数学课程成绩统计结果如下：

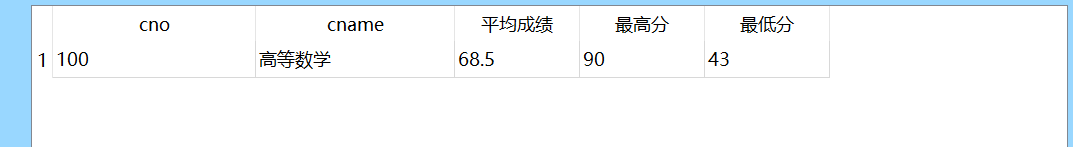


图 28 课程成绩统计结果

成绩区间统计界面：

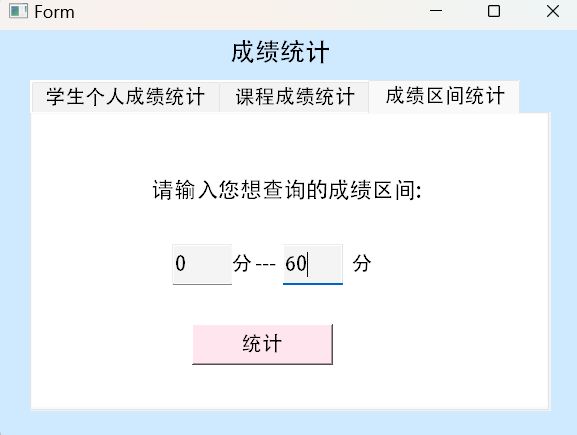


图 29 成绩统计界面成绩区间统计

成绩在0-60分区间的同学统计结果如下：



图 30 成绩区间统计结果

3.8课程操作界面

所有课程信息如下：



图 31 显示所有课程的信息界面

课程查询：

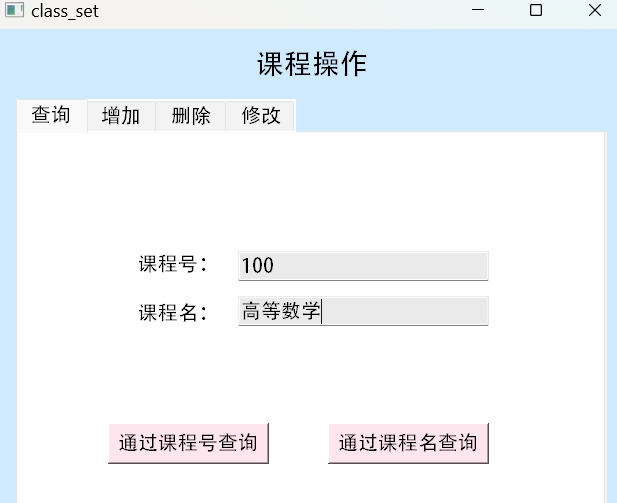


图 31 课程信息查询界面

高等数学课程的查询结果如下：

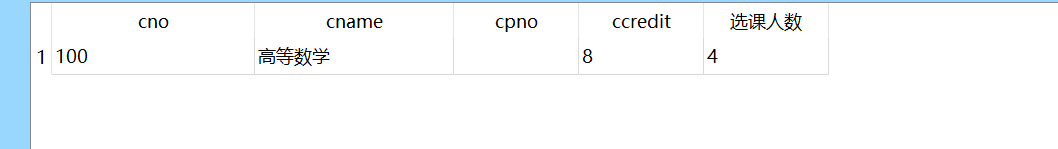


图 32 课程信息查询结果

课程增加界面：

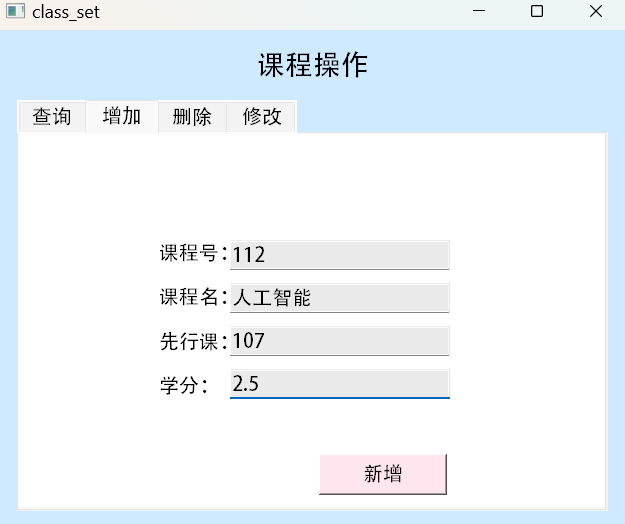


图 33 课程操作界面增加部分

增加人工智能课程的结果如下：



图 34 课程增加结果

删除课程界面：



图 35 课程操作界面删除部分

课程号为111的模式识别课程删除后结果如下：

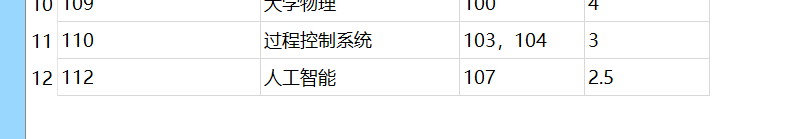


图 36 课程删除结果

课程修改界面：

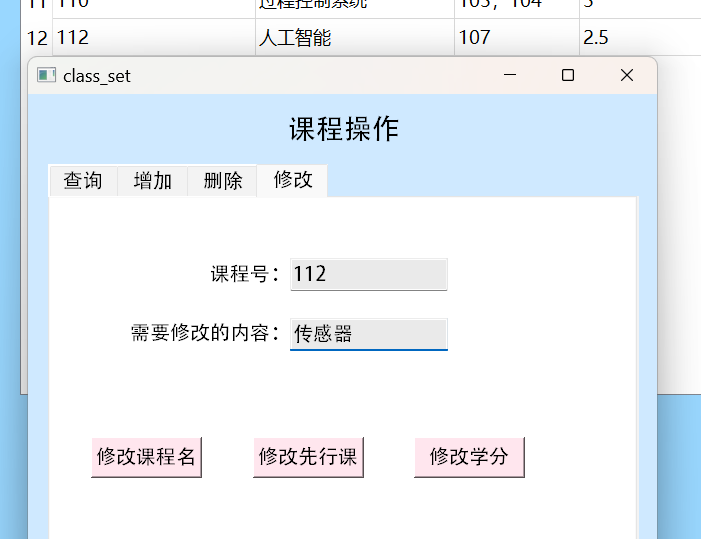


图 37 课程操作界面修改部分

把课程号为112的课程名称修改为传感器的结果如下：



图 38 课程操作界面修改结果

## 四、实验总结

我认为本部分是本次数据库实验中最重要的一部分，在这部分实验中，我利用Qt和SQLite构建了自己的学生信息管理系统。这部分的实验安排很好的联系了另外的科目，在学习数据库的同时温习了Qt的使用，使我受益良多。

在完成实验的过程中我也遇到了非常多的问题，例如无法正常连接数据库、connect无法正常连接信号与槽函数、子ui无法和主ui正常连接、字符串无法正常拼接等。通过解决各种问题、调试各类bug使我对SQL语言有了更深入的了解，同时编程能力和调试能力也有了较大的提升。

由于期末时间紧凑，无法全身心的投入本次实验，在本次实验中所写的代码冗余度大、可读性差。在寒假假期时间充裕时，我会将界面再次优化、尝试新的数据库模型、尝试使用全局变量来优化算法等。

## 五、问题解决

### 5.1 问题一

问题描述：connect函数无法正常连接

问题原因：Qt版本不兼容

解决方法：信号与槽通过connect函数连接有多种写法，其中最老的一种写法如下图所示。这种写法的优点是清楚明了，但是当信号函数和槽函数出现错误时并不会报错，这对于调试程序非常不友好，虽说市面上主流的Qt教程均采用这种写法，但是现在部分版本的Qt已经不再兼容这种写法，所以在使用connect函数是建议采用以下这种新写法。



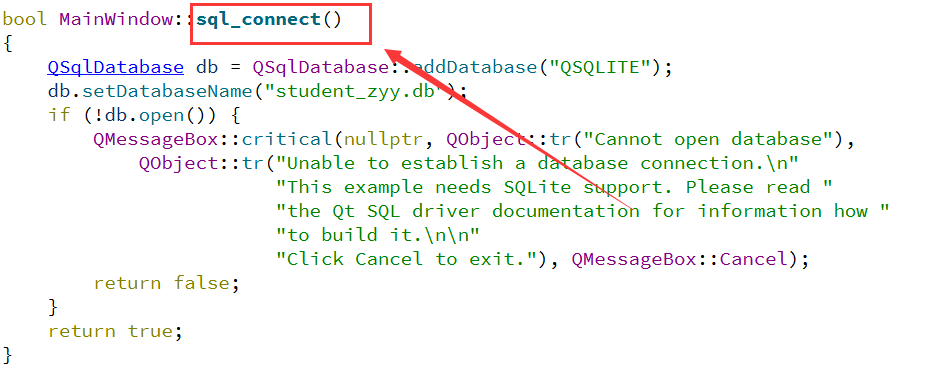
另外需要注意的是，connect函数中信号函数和槽函数不需要带形参，这也是使用connect函数出错的主要原因之一。

### 5.2 问题二

问题描述：无法正常连接数据库

问题原因：连接函数与Qt内置函数connect重名导致不兼容

解决方法：更改连接函数的名称即可，如下图将函数命名为sql\_connect()即可。



### 5.3 问题三

问题描述：需要拼接多个字符串时addBindValue()函数无法正常使用

解决方法：拼接多个字符串理论上是可以通过在prepare()函数后不断加入addBindValue()函数来实现的，但是在实际操作时并不能正常使用。解决方法是可以通过bindValue()将多个字符串拼接起来，如下图所示。

